Global Contour Lines Reconstruction in Topographic Maps

Joachim Pouderoux, Salvatore Spinello

Global Contour Lines Reconstruction in Topographic Maps

Joachim Pouderoux Salvatore Spinello
Iparla Project (LaBRI - INRIA)
University of Bordeaux, France
pouderou@labri.fr spinello@inria.fr

Abstract

Topographic maps are a common support for geographical information because they have the particularity to portray the relief through a set of contour lines. This topographic feature can be very useful in many context but the automatic extraction of this information is not an easy task, especially because the map contains many other layers which overlay the contours. In this paper we propose an automatic approach to reconstruct gaps in contour lines. Our novel parameterless reconstruction scheme is based on the extrapolation of the gradient orientation field from the available pieces of thinned contours. A weight is then affected to each pair of end-points according the force needed by its potential reconstructed curve to cross the field. The computation of the optimal global solution is obtained by solving a perfect matching problem. We finally use the orientation flow to fill the gaps with a smooth curve that respects the tangents at the en-points.

Figure 1. Sample of the topographic map of lake Winnibigoshish, Minnesota.

1 Introduction

With the evolution of data acquisition, storage techniques and the growth of planet-wide networks accessible through a large amount of different devices, Geographical Information Systems (GIS)-based applications are more and more used.

While modern *topographic maps* (also known as *contour maps*, ie. maps describing the topology of a part of the earth) are issued from GIS, large libraries of paper maps (of recent or older ages) are still available and provide a low-cost alternative to expensive and incomplete remote sensing databases. Thus, the digital acquisition of these maps becomes necessary for example to analyze and use the represented terrains with today's tools.

The most interesting data contained on topographic maps are obviously the contours lines, which are imaginary lines that join points of equal elevation on the surface above or below a reference surface such as the mean sea level. However, traditional topographic maps show much more than a contour map layer: it can portray rivers, roads, buildings, forest areas, etc. An issue for an automatic extraction of contours is that these other symbols overlay the contours layer. Thanks to the specific contour lines colors, the extraction of the contour lines layer is made easy. This process, unfortunately, produces gaps in the contours lines. While previous work in the area provide local and non optimal solutions, we introduce a novel global approach to fill these gaps based on the orientation field generated from the available pieces of thinned contours. Our technique tends to imitate the *good continuation* law of Gestalt which characterizes human vision perception. This principle states that graphic elements that suggest a continued visual line will tend to be grouped together in our mind. In our approach we will also consider the principle that a topographical contour line is expected to be almost parallel to its adjacent contours.

This paper is organized as follows: In a first part we analyze previous approaches that have been proposed for the general problem of contour lines vectorization from topo-

graphic maps. In section 3 we present the 3 steps of our reconstruction technique. Finally, we quickly present our results before to conclude.

2 Related work

Automated recognition of topographic map has been going on for many years resulting in a huge amount of publications. Early reports about the vectorization of line drawings already introduced the main necessary steps of any automatic procedure: i) digitalization of the original paper document; ii) filtering; iii) thresholding; iv) thinning and pruning the binary image; v) raster to vector conversion. These steps can be found in [14] for the automatic vectorization of clean contour and drainage/ridge sheets, in [11] for an early attempt to extract elevation contour lines on topographical maps. In this paper, we focus on task v) and especially on the problem of gap filling. Therefore, in this section we point out only references to this specific problem in the raster to vector procedure.

The image based approach. The most common approach for the raster to vector conversion are strictly based on perceptual principles. Indeed, to decide for closing or grouping two different segments/pixels, the main two criteria used are proximity and continuity.

In [8], a raster to vector technique is presented to process paper-based maps. The problem of gap reconstruction is solved by assuming that there is only one possible continuation from an end-point. The natural continuation can be found along the current direction of the line. The gap is crossed by searching from the point at the end of the line within a sector around the current direction. This approach was recently used in [3]. In [2], the 5 steps mentioned earlier are used to reconstruct contour lines from color topographical maps using a techniques based on mathematical morphology. A combination of Euclidean distances between extremities and differences between their tangential directions is used to join the disconnected lines in a very locally fashion. This latest approach is also used in [12] coupled to a A* search algorithm.

Albeit attractive because of their simplicity, all the existing closing algorithm based on perception criteria fail (see figure 2).

The geometric based approach. The curve reconstruction problem can be analyzed as an instance of the more general problem: given a finite sample V of an unknown curve λ, the task is to construct a graph $G = (V, E)$ in such a way that two points in V are connected by an edge of G iff. the points are adjacent on λ. The graph G is called a polygonal reconstruction of λ. The curve reconstruction

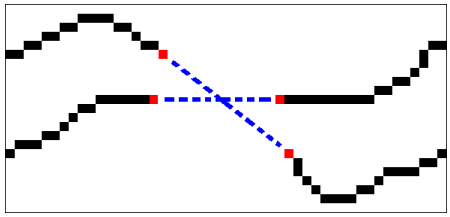

Figure 2. Problem of image based reconstruction.

problem has received a lot of attention in the graphics and the computational geometry community and a great amount of work has been written. The first algorithms for curve reconstruction imposed a uniform sampling condition, as they basically demanded that the distance between any two adjacent samples must be less than a given constant. This is not satisfactory as it may require a dense sampling in areas where a sparse sampling is sufficient.

[1] introduced the concept of the local feature size (distance of a point to the medial axis of his curve). Using this concept they define a non-uniform sampling condition that allows for sampling of variable density. Then they give an algorithm that, from a sample set of a collection of smooth closed curves, which satisfies this sampling condition, computes the correct reconstruction. This algorithm works by computing the Delaunay Triangulation of the point set and then filtering it to obtain the reconstruction. [5] extended this work to handle a collection of open and closed smooth curves also using Delaunay filtering. [6] gaves an algorithm that allegedly handles well corners and endpoints. The algorithm has no guarantee and, in fact, it is not difficult to find counterexamples where it fails. Recently, in [15], authors propose to vectorize the contour lines using a Delaunay triangulation where the Delaunay edges are filtered using both local and global rules. However, the global approach they used is not optimal while it use a greedy algorithm and the connecting rules are strictly geometric.

The gradient vector flow approach. As say, our method is based on the gradient orientation field generated by the input contour lines. However, orientation field has already been studied in image analysis. For example, [4] present how an orientation interpolation operator can be used to recover geometrical information in images. Another application to snakes is given in [10].

3 Contours reconstruction

Overview In this paper, we don't address the specific problem of contour lines extraction from color or grayscale maps. We assume that this step as already been performed.

The presented technique requires a binary map made of spaghetti contours.

The basic idea of our technique is to reconstruct the global gradient orientation field of the available contours lines and to use it in a globally way to find the natural pairs of end-points which should be reconnected. Once matching has been obtained we also use the orientation field to ensures a smooth reconstruction.

The workflow of our algorithm can be summarized as follows: i) gradient orientation field generation from contour line normals; ii) matching end-points; iii) fill the gap between each pair of end-points.

3.1 Orientation field generation

The first step of our algorithm consists in a orientation field generation over the full map starting from the gradients computed over the input contour lines. Actually, it is very important to understand that we are not interested in vector direction, we only rely on vector orientation. The field F is a $m \times n$ array (size of the input image) of real angle values between $]-\pi, \pi[$.

This field is the key of our method therefore we put special attention for this task. The orientation field computation is achieved by: i) computing the orientation of normal vectors on each pixel of the given contour lines; ii) interpolating the orientation values on background pixels.

3.1.1 Computation of contours' normal

It is not that obvious to compute normal vectors for each pixel on a discrete curve. Some very specific methods for 2D tangent estimation are studied in the field of discrete geometry (most efficient techniques are compared in [13]), however after some tests, the computed tangent were not smooth enough for our needs. We rather made the choice to interpolate each contour line with a B-spline and then, evaluate the normal vector accurately. As said before, we are only interested in the vector orientation but not in the vector direction.

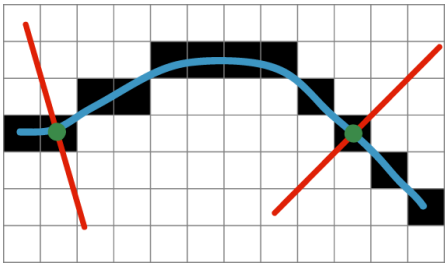

Figure 3. Orientation of normals on B-spline interpolating a contour.

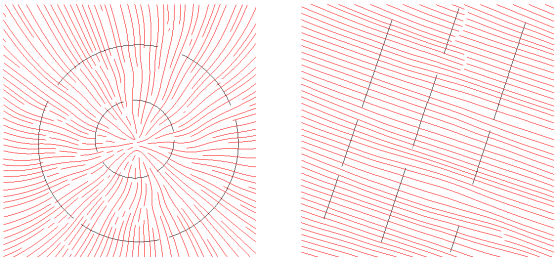

Figure 4. Streamlines of the reconstructed orientation field for 2 cases of study.

The B-spline interpolation of contours can be seen as a raster to vector operation. Because contours in the image are previously thinned to a 8-connected component, this operation is an easy task. We evaluate the B-spline derivative along the contour line using the de Boor's algorithm and calculate the normal vectors (see figure 3). The angle of the normal vectors are stored in the field F as orientation.

3.1.2 Field interpolation

Once each normal orientation is known on the input contour lines, we can proceed to a field interpolation. Our goal is to estimate the orientation of the normals at every point of the image. Such an interpolation is traditionally solved by solving a partial differential equation (PDE). For example, an AMLE interpolation operator is presented in [4]. However, we rather propose a faster implementation based on a front propagation of known values. The function Θ is a key function which computes the mean orientation of a point p using a 3×3 window in the 8-neighborhood $N(p)$ of p:

$$\Theta(p) = \frac{\sum_{q \in N(p)} \Lambda(F(q), F(p))}{|N(p)|} \qquad (1)$$

The function Λ ensures the consistency of the orientation operations between two angles α and $beta$ in F:

$$\Lambda(\alpha, \beta) = \begin{cases} \alpha & if d1 = min(d1, d2, d3) \\ \alpha + \pi & if d2 = min(d1, d2, d3) \\ \alpha - \pi & if d3 = min(d1, d2, d3) \end{cases} \qquad (2)$$

with $d1 = |\beta - \alpha|$, $d2 = |\beta - \alpha - \pi|$ and $d3 = |\beta - \alpha + \pi|$.

Figure 4 shows the stream lines of the orientation fields obtained for 2 simple sets of contours (circular and linear).

3.2 Connecting end-points

To fill gaps in contour lines we always have to match two end-points. Similar problems are well known in graph

theory as the *Matching problem*. In this part, we determine pairs of contour extremities which should be connected together. This is done in two steps: i) computing the energy needed to go from one end-point to the other through the orientation field; ii) solving the perfect matching problem.

3.2.1 Matching's weight estimation

For each pair of end-points, we associate a weight given by an energy function. This function must be defined in such way that: i) the energy is zero if the end-points directly meet together by following the flux line, ii) the energy is maximal proportionally to the path length if the path to join them is everywhere normal to the gradient orientation field. The weight function is calculated as follows:

$$\omega(e_1, e_2) = \sum_{t=0}^{1} \left| \overrightarrow{p-q} \cdot \overrightarrow{F(p)} \right| . \qquad (3)$$

The matrix W is constructed by calculating function ω for each pair of end-points. Figure 5 illustrates the reconstruction scheme: in black are the original contour lines C_1 and C_2, in light gray the contour lines issued from the end-points of e_1 and e_2, and, in blue are the pixels of the curve reconstructed using the algorithm described in 3.3.

3.2.2 Perfect matching

Once the matrix W has been computed, we proceed to the global matching of the end-points. Let $G(V, E)$ be a graph and $w : E \rightarrow \mathbb{R}$ the cost function, for example the function 3. A *perfect matching* of G is a subset $M \subset E$ such that every vertex of G is incident to exactly one edge of M. The weight $w(M)$ of a matching M is the sum of the weights of its edges. The problem is to find a perfect matching of maximum weight. This problem can be solved in polynomial time by the algorithm of Edmonds [7]. We solve the matching using the implementation in $O(N^3)$ based on the Gabow's work [9].

3.3 Smooth gap reconstruction

Once the matching obtained, we have to fill the gap between these end-points. Several methods exist: the simplest one is of course to draw a line segment, but it is obvious that, in most cases, it gives a bad looking result and even it can cause topological errors (see figure 6). A better result can be obtained by drawing a Bezier curve or even better a Hermite interpolation. However, we just have tangents' orientation at the end-points, therefore, we had to find a suitable interpolation process that fills the gaps obtaining a smooth curve with tangents $\overrightarrow{t_1}$ and $\overrightarrow{t_2}$ at the end-points (see figure 5).

For each contour line end-point, we store the contour lines (in this case, the points obtained by integration of the gradient of the gradient field) that is the points which naturally continue the contour line in the reconstructed field (gray curves in figure 5).

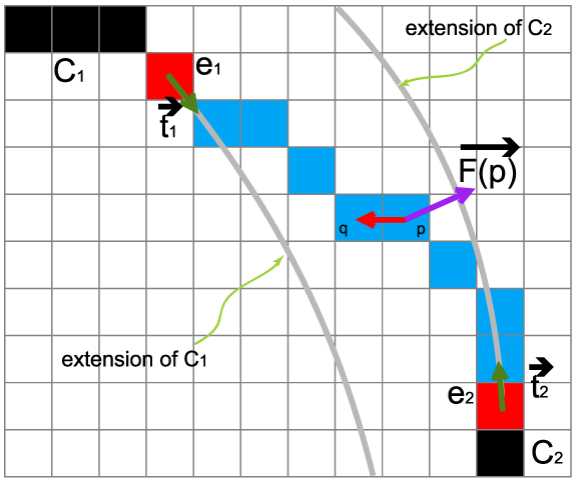

Figure 5. Reconstructed contour between a pair of end-points.

As said, the basic idea is to linearly interpolate the flow field based contours issued from each pair of end-point e_1 and e_2. The points are stored in 2 queues Q_{e_1} and Q_{e_2} of respective size $S(Q_{e_1})$ and $S(Q_{e_2})$. The coordinates of such points are parameterized as follows:

$$p(t) = \begin{pmatrix} x_{p_{e_1}(t)} * (1-t) + x_{p_{e_2}(t)} * t \\ y_{p_{e_1}(t)} * (1-t) + y_{p_{e_2}(t)} * t \end{pmatrix} \qquad (4)$$

where t takes its value in the interval $[0, 1]$, $p_{e_1}(t) = Q_{e_1}[S(Q_{e_1}) * t/d]$, $p_{e_2}(t) = Q_{e_2}[S(Q_{e_2}) - (S(Q_2) * t/d)]$. $Q_{e_1}[x]$ denotes the x^{th} point in the queue Q_{e_1} and d is the Euclidean distance between the two end-points.

4 Results

Figure 8 depicts the reconstructed contours obtained with our method. The reconstructed field is shown in figure 7. The size of the map is 1278×903 pixels. On a

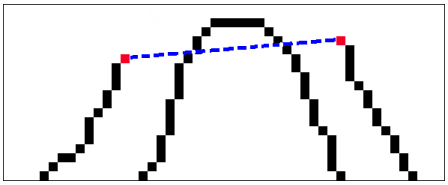

Figure 6. Problem of image based reconstruction.

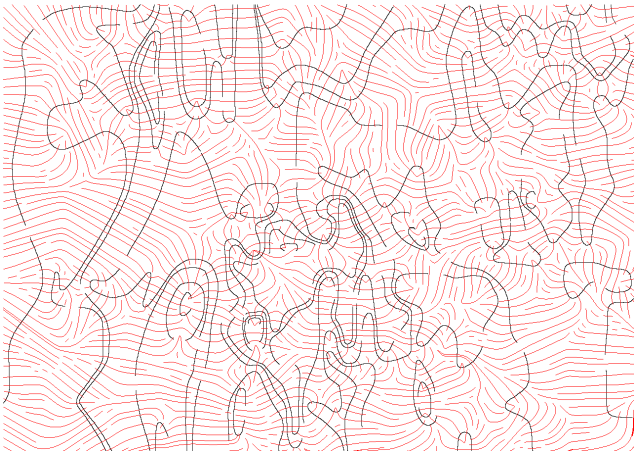

Figure 7. Streamlines of the reconstructed orientation field from the input contours.

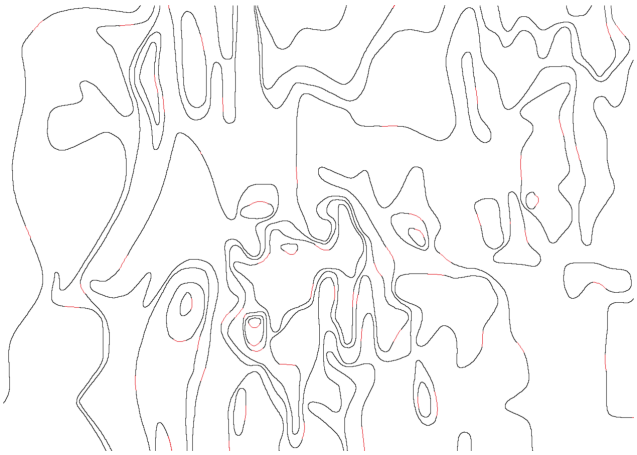

Figure 8. The reconstructed contour map.

Core 2 Duo at 1.86GHz, the total computing time is about 23 seconds: 22s for orientation field interpolation and less than 1s for the matching and reconstruction steps. We can see that 100% of the gaps were correctly restored and the reconstructed parts are smooth and look natural. We have performed our technique on several other maps with comparable excellent results. In the worst case, there were one or two errors.

5 Conclusion

We have presented an elegant, parameterless and efficient technique to reconstruct broken contour lines. Our method is based on the gradient orientation field of input contours. The global aspect of the reconstruction is based on both the orientation field and the perfect matching performed on all potentially matching end-points. The obtained results on different maps are very satisfying in both term of error rate and look.

Our method has been applied to reconstruct topographical contour lines but it could be applied in many other applications in document analysis and computer vision domains.

References

[1] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing*, 60(2):125–135, 1998.

[2] P. Arrighi and P. Soille. From scanned topographic maps to digital elevation models. In *Proceedings of Geovision'99*, 1999.

[3] Y. Chen, R. Wang, and J. Qian. Extracting contour lines from common-conditioned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing*, 44(4):1048–1057, 2006.

[4] A. Chessel, R. Fablet, F. Cao, and C. Kervrann. Orientation interpolation and applications. In *Proceedings of IEEE International Confernece on Image Processing: ICIP'06*, October 2006.

[5] T. K. Dey, E. A. Ramos, and K. Mehlhorn. Curve reconstruction: connecting dots with good reason. In *Proceedings of the 15th Symposium on Computational Geometry*, pages 197–206, 1999.

[6] T. K. Dey and R. Wenger. Reconstruction curves with sharp corners. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 233–241, 2000.

[7] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–667, 1965.

[8] L. Eikvil, K. Aas, and H. Koren. Tools for interactive map conversion and vectorization. *Third International Conference on Document Analysis and Recognition: ICDAR 1995*, 02:927, 1995.

[9] H. N. Gabow. *Implementation of algorithms for maximum matching on nonbipartite graphs.* PhD thesis, Stanford University, 1974.

[10] D. Gil and P. Radeva. Extending anisotropic operators to recover smooth shapes. *Computer Vision and Image Understanding*, 99(1):110–125, 2005.

[11] D. Greenlee. Raster and vector processing for scanned line work. *Photogrammetric Engineering and Remote Sensing*, 53(10):1383–1387, 1987.

[12] A. Khotanzad and E. Zink. Contour line and geographic feature extraction from usgs color topographical paper maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):18–31, 2003.

[13] J.-O. Lachaud, A. Vialard, and F. de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In *Proceedings of Int. Conf. Discrete Geometry for Computer Imagery (DGCI'2005)*, pages 140–251, 2005.

[14] F. Leberl and D. Olson. Raster scanning for operational digitizing of graphical data. *Photogrammetric Engineering and Remote Sensing*, 48(4):615–627, 1982.

[15] S. Spinello and P. Guitton. Contour line recognition from scanned topographic maps. In *Proceedings of Winter School of Computer Graphics: WSCG*, pages 419–426, 2004.